

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In Re Application of:

Date: March 30, 2007

Philip S. COULTHARD, et al.

Confirmation No: 3395

Serial No: 09/879,024

Group Art Unit: 2178

Filed: June 11, 2001

Examiner: Kyle R. Stork

Title: METHOD AND SYSTEM FOR CONVERTING USER INTERFACE
SOURCE CODE OF A LEGACY APPLICATION TO WEB PAGES

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

BRIEF ON APPEAL

(1) Real Party in Interest

The real party in interest is International Business Machines Corporation by virtue of an assignment from the inventors recorded in the U.S. Patent Office on October 2, 2001, reel no. 012570, frame no. 0628.

(2) Related Appeals and Interferences

There are no appeals, interferences, or judicial proceedings known to Appellant, the Appellant's legal representative, or Assignee, which may be related to, directly affect, be directly affected by, or have a bearing on the decision by the Board of Patent Appeals and Interferences in the pending appeal.

(3) Status of Claims

Claims 1-14 and 16 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Application Pub. No. 2002/0120787 (“Shapiro”) in view of U.S. Patent Application Pub. No. 2004/0015839 (“Sarkar”).

Claim 15 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Shapiro and Sarkar, in further view of U.S. Patent No. 6,721,713 (“Guheen”).

Claims 1-16 are being appealed.

(4) Status of Amendments

There are no unentered amendments.

(5) Summary of Claimed Subject Matter

Distributed computing is a paradigm of computing in which clients and servers are interconnected through a network (e.g., the Internet). In such a paradigm, software applications used by a given client typically reside on a server. Specification, page 1, lines 10-15. Distributed computing has changed the methodology for writing code of software applications that reside on a server – e.g., rather than writing one massive self-contained application having thousands of lines of code, software developers now typically select and weave together smaller portions of code, called components. Each component generally has a particular function that can be reused across multiple applications on the Internet. Specification, page 1, line 20 – page 2, line 4.

Self-contained legacy applications having thousands of lines of code, however, may still be very important to businesses and organizations. The term “legacy applications” refers to existing applications which remain in operation with a business or

organization but were developed before the advent of distributed computing and structured programming. Specification, page 3, line 20 – page 4, line 3. Interactive legacy applications that have mixed business and presentation logic must be restructured to be enabled as web applications. For example, some legacy applications have thousands of different user interface screens, and extraordinary effort and time is generally required to convert the user interface display files to existing web pages. Specification, page 4, lines 7-13. The present invention is directed to techniques for converting display pages of a legacy application to modern Web browser user interfaces. Specification, page 5, lines 9-10.

Accordingly, independent claim 1 recites a method for converting display source code of a legacy application having mixed business and presentation logic on a server to a network interactive web-browser page. The method includes resolving the display source code of the legacy application into a plurality of record formats, in which each record format corresponds to source code associated with an input/output screen of the legacy application. Specification, page 17, lines 18-21; page 8, lines 8-9. The method further includes parsing each record format into a corresponding intermediate file that is renderable by a web browser, in which each intermediate file including static content and dynamic content. The static content corresponds to an unchanging portion of a given input/output screen of the legacy application, and the dynamic content corresponds to a dynamic portion of the given input/output screen that is filled in at runtime by the legacy application. Specification, page 17, line 22 – page 18. The method further includes converting the static content of each intermediate file to a corresponding web page for display on the web browser including creating dynamic components for populating the

web page based on the dynamic content of the intermediate file. Specification, page 19, lines 7-19.

Independent claim 10 recites a computer readable medium containing program instructions tangibly stored thereon for creating web interfaces of a legacy application having mixed business and presentation logic stored on a computer. The computer readable medium comprises instructions for parsing display file data description source of the legacy application to render the display file data description source into a plurality of intermediate files that are each renderable by a web browser of a client, in which each intermediate file corresponds to a record format representing source code associated with an input/output screen of the legacy application. Specification, page 17, lines 18-21; page 8, lines 8-9. The computer readable medium further comprises instructions for converting each intermediate file to a corresponding web page, in which at least one data object maintains input data, output data, and feedback data of the legacy application on the client, and at least one web page maintains static content of a given input/output screen of the legacy application. Specification, page 19, lines 7-19. The computer readable medium further comprises instructions for dynamically updating the at least one web page with the input data, output data, and feedback data of the legacy application via a servlet instance. Specification, page 16, lines 13-14. The computer readable medium further comprises instructions for displaying the dynamically updated web page through the web browser on the client via a network. Specification, page 17, lines 8-10.

Independent claim 11 recites a computer readable medium containing program instructions tangibly stored thereon for use in a computer network. The computer readable medium comprises instructions for providing a plurality of intermediate files that

are renderable by a web browser, in which each intermediate file corresponds to a record format representing source code associated with an input/output screen of a legacy application and the legacy application has mixed business and presentation logic. Specification, page 19, lines 7-19. The computer readable medium further comprises instructions for converting each intermediate file to a corresponding web page, in which a static portion of a given web page corresponds to a static portion of the corresponding record format and a dynamic portion of the given web page interacts with display input data, output data, and feedback data required by and from the legacy application. Specification, page 19, lines 7-19; page 20, lines 4-6.

Independent claim 12 recites a computer system for executing an application. The computer system includes a central processing unit, a main memory connected to the central processing unit with a communication bus, and a data storage unit connected to a data storage interface that is connected to the communication bus. Specification, page 9, lines 12-15. The computer system further includes at least one input/output device connected to the communication bus and connected to a network interface to an external computer network. Specification, page 12, lines 2-10. The computer system further includes a legacy application having mixed business and presentation logic stored in the main memory and capable of executing on the central processing unit. Specification, page 11, lines 2-4. The computer system further includes a plurality of intermediate files that are renderable by a web browser, in which each intermediate file corresponds to a record format representing source code associated with an input/output screen of the legacy application. Specification, page 17, lines 18-21; page 8, lines 8-9. As the legacy application executes, application logic uses either a legacy application display of

associated with a given record format or the plurality of intermediate files for communication of the legacy application to a user over the external computer network. Specification, page 8, lines 1-4; page 17, lines 10-12.

Independent claim 13 recites a computer server for converting the display source of a legacy application having mixed business and presentation logic stored and executing on a computer. The computer server includes a central processing unit. Specification, page 11, lines 15-17. The computer server further includes a parser to parse the display source of the legacy application into a plurality of record formats, in which each record format is unique to each input/output screen definition of the legacy application. Specification, page 17, lines 18-21; page 8, lines 8-9. The computer server further includes a generator of intermediate files having nested tags of each of the record formats, in which each intermediate file is renderable by a web browser. Specification, page 18, line 22 – page 19, line 1. The computer server further includes a converter operable to convert the intermediate files to one or more web pages. Specification, page 19, lines 7-9. The converter includes an object creator to create dynamic components for dynamic portions of the record formats, in which the dynamic components populate the one or more web pages. The object creator further creates a static component for display of unchanging aspects of the record formats, in which the static component represents the one or more web pages. Specification, page 19, lines 7-19.

(6) Grounds of Rejection to be Reviewed on Appeal

Appellant requests review as to claims 1-14 and 16 and their rejection under 35 U.S.C. § 103(a) as being unpatentable over Shapiro in view of Sarkar.

(7) **Argument**

1. **Claims 1-14 and 16 are not properly rejected under 35 U.S.C. § 103(a) as being unpatentable over Shapiro in view of Sarkar.**

(A) **Claims 1-14 and 16**

Claim 1 recites a method for converting display source code of a legacy application having mixed application logic and presentation logic on a server to a network interactive web-browser page.

In particular, the method includes resolving the display source code of the legacy application into a plurality of record formats, in which each record format corresponds to source code associated with an input/output screen of the legacy application. The method further includes parsing each record format into a corresponding intermediate file that is renderable by a web browser, in which each intermediate file includes static content and dynamic content. The static content corresponds to an unchanging portion of a given input/output screen of the legacy application, and the dynamic content corresponds to a dynamic portion of the given input/output screen that is filled in at runtime by the legacy application. Each intermediate file is then converted into a web page.

A. Shapiro Fails To Disclose Resolving Display Source Code of a Legacy Application Into a Plurality of Record Formats, in Which Each Record Format Corresponds To Source Code Associated with an Input/Output Screen of the Legacy Application

Shapiro discloses a system and method for accessing functionality of a backend system from an application server (see Abstract). The backend system may be a legacy system that implements one or more callable functions (paragraphs 0034-0035). To

provide access to the legacy system through an application server, Shapiro's system includes a data mining (client) computer system 82 that executes software code on the backend system 112 and receives information specifying the functionality of the backend system 112. The data mining client 82 then analyzes the received information and programmatically creates information based on the analysis for accessing the functionality of the backend server (paragraph 0090).

Accordingly, Shapiro discloses programmatically creating information that is useable for accessing the functionality of the backend system based on analysis of one or more callable functions of a legacy system (see also paragraph 0039). Thus, in Shapiro's system there is no teaching or suggestion (or a need) for resolving display source code of a legacy application into a plurality of record formats, in which each record format corresponds to source code associated with an input/output screen of the legacy application, as Shapiro's system requires an analysis of callable functions.

In contrast, claim 1 recites resolving display source code of a legacy application into a plurality of record formats, in which each record format corresponds to source code associated with an input/output screen of the legacy application. Each record format is then parsed into an intermediate file and converted into a corresponding web page. Claim 1, therefore, requires generating web pages based on each input/output screen of the legacy application.

Shapiro and claim 1 clearly use two different techniques for accessing legacy applications – i.e., Shapiro accesses a legacy system based on callable functions of the legacy system, whereas claim 1 requires creation of web pages based on input/output screens of a legacy application.

B. Sarkar and Guheen Fail To Disclose Resolving Display Source Code of a Legacy Application Into a Plurality of Record Formats, in Which Each Record Format Corresponds To Source Code Associated with an Input/Output Screen of the Legacy Application

Sarkar discloses a method and system for running application code originally developed as simple Java Beans, in an Enterprise Java Bean (EJB) environment, without modifying the original application code (see Abstract). Guheen discloses a method and system for identifying alliances among a plurality of business entities in components of a network framework (see Abstract). The Examiner does not cite Sarkar or Guheen as disclosing resolving display source code of a legacy application into a plurality of record formats, in which each record format corresponds to source code associated with an input/output screen of the legacy application. Nevertheless, as with Shapiro, Sarkar and Guheen each fail to disclose this limitation.

C. The claim has limitations not taught by either reference

To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974).

Shapiro, Sarkar and Guheen fail to disclose resolving display source code of a legacy application into a plurality of record formats, in which each record format corresponds to source code associated with an input/output screen of the legacy application. Consequently, the combination of Shapiro, Sarkar and Guheen cannot render claim 1 obvious.

Claims 2-9 and 14-15 depend from claim 1 and are allowable over Shapiro, Sarkar and Guheen for reasons corresponding to those set forth in connection with claim 1.

D. Other Independent Claims

Independent claims 10-13 incorporate limitations similar to those of claim 1. Claims 10-13 (and the claims that depend therefrom) are also allowable over the combination of Shapiro, Sarkar and Guheen for reasons corresponding to those set forth with respect to claim 1.

Conclusion

Neither Shapiro, Sarkar nor Guheen discloses resolving display source code of a legacy application into a plurality of record formats, in which each record format corresponds to source code associated with an input/output screen of the legacy application, as required by the claims. Appellant, therefore, respectfully submits that claims 1-16 are not properly rejected under § 103.

Please charge any fee that may be necessary for the continued pendency of this application to Deposit Account No. 09-0460 (IBM Corporation).

Respectfully submitted,
SAWYER LAW GROUP LLP



Kelvin M. Vivian
Attorney for Appellant
Reg. No. 53,727
(650) 475-1448

March 30, 2007

Date

Appendix of Claims

1. (Previously Presented) A method for converting display source code of a legacy application having mixed business and presentation logic on a server to a network interactive web-browser page, the method comprising:

resolving the display source code of the legacy application into a plurality of record formats, each record format corresponding to source code associated with an input/output screen of the legacy application;

parsing each record format into a corresponding intermediate file that is renderable by a web browser, each intermediate file including static content and dynamic content, the static content corresponding to an unchanging portion of a given input/output screen of the legacy application, the dynamic content corresponding to a dynamic portion of the given input/output screen that is filled in at runtime by the legacy application; and

converting the static content of each intermediate file to a corresponding web page for display on the web browser including creating dynamic components for populating the web page based on the dynamic content of the intermediate file.

2. (Previously Presented) The method of claim 1, wherein each web page is displayed on the Internet.

3. (Previously Presented) The method of claim 1, wherein each web page is displayed on a network selected from the group consisting of: an internal network, an

Intranet, a LAN, a WAN, an internal bus, a wireless network.

4. (Previously Presented) The method of claim 1, wherein each intermediate file is an XML language file.

5. (Original) The method of claim 4, wherein the XML language file is an HTML file.

6. (Original) The method of claim 4, wherein the XML language file is a WML file.

7. (Previously Presented) The method of claim 1, wherein each web page comprises a JavaServer Page.

8. (Original) The method of claim 1, wherein the dynamic components further comprise JavaBeans.

9. (Previously Presented) The method of claim 2, wherein each web page is stored on the server.

10. (Previously Presented) A computer readable medium containing program instructions tangibly stored thereon for creating web interfaces of a legacy application having mixed business and presentation logic stored on a computer, the program instructions for:

parsing display file data description source of the legacy application to render the display file data description source into a plurality of intermediate files that are each renderable by a web browser of a client, each intermediate file corresponding to a record format representing source code associated with an input/output screen of the legacy application;

converting each intermediate file to a corresponding web page, wherein at least one data object maintains input data, output data, and feedback data of the legacy application on the client, and at least one web page maintains static content of a given input/output screen of the legacy application;

dynamically updating the at least one web page with the input data, output data, and feedback data of the legacy application via a servlet instance; and

displaying the dynamically updated web page through the web browser on the client via a network.

11. (Previously Presented) A computer readable medium containing program instructions tangibly stored thereon for use in a computer network, the computer readable medium containing program instructions for:

providing a plurality of intermediate files that are renderable by a web browser, each intermediate file corresponding to record format representing source code associated with an input/output screen of a legacy application, the legacy application having mixed business and presentation logic; and

converting each intermediate file to a corresponding web page, a static portion of a given web page corresponding to a static portion of the corresponding record format and

a dynamic portion of the given web page interacting with display input data, output data, and feedback data required by and from the legacy application.

12. (Previously Presented) A computer system for executing an application, comprising:

a central processing unit;

a main memory connected to the central processing unit with a communication bus;

a data storage unit connected to a data storage interface which is connected to the communication bus;

at least one input/output device connected to the communication bus and connected to a network interface to an external computer network,

a legacy application having mixed business and presentation logic stored in the main memory and capable of executing on the central processing unit; and

a plurality of intermediate files that are renderable by a web browser, each intermediate file corresponding to a record format representing source code associated with an input/output screen of the legacy application;

wherein as the legacy application executes, application logic uses either a legacy application display of associated with a given record format or the plurality of intermediate files for communication of the legacy application to a user over the external computer network.

13. (Previously Presented) A computer server for converting the display source of a

legacy application having mixed business and presentation logic stored and executing on a computer, comprising:

- a central processing unit;

- a parser to parse the display source of the legacy application into a plurality of record formats, each of the record formats being unique to each input/output screen definition of the legacy application;

- a generator of intermediate files having nested tags of each of the record formats, each intermediate file being renderable by a web browser; and

- a converter operable to convert the intermediate files to one or more web pages, the converter further comprising:

 - an object creator to create dynamic components for dynamic portions of the record formats, the dynamic components for populating the one or more web pages; and

 - a static component for display of unchanging aspects of the record formats, the static component representing the one or more web pages.

14. (Previously Presented) The method of claim 1, wherein each step of the method occurs at development time during which a user is preparing a new user interface for the legacy application.

15. (Previously Presented) The method of claim 14, wherein converting the static content of each intermediate file is performed offline without any remote connection to the server upon which the legacy application resides.

16. (Previously Presented) The computer readable medium of claim 10, wherein:
- the program instructions for parsing the display file data description source and converting each intermediate file are each executed during development time of the web-browser page; and
- the program instructions for dynamically updating the at least one web page and displaying the dynamically updated web page are each executed during runtime.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None